Diamond Star Pattern

Given an integer **N,** print the following star diamond pattern. Print "pattern not possible" if pattern can not be printed.

Example:

```
Input: N = 3
Output:
   *
  ***
 *****
 *****
  ***
   *


Input: N = 1
Output:
*
*


Constraints :
1   <= N <= 20
```

C Code:

```c
#include <stdio.h>
void erect_pyramid(int N)
{
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N - i - 1; j++)
        {
            printf(" ");
        }
```

```c
        for (int j = 0; j < 2 * i + 1; j++)

        {

            printf("*");

        }

        for (int j = 0; j < N - i - 1; j++)

        {

            printf(" ");

        }

        printf("\n");

    }

}


void inverted_pyramid(int N)

{

    for (int i = 0; i < N; i++)

    {

        for (int j = 0; j < i; j++)

        {

            printf(" ");

        }

        for (int j = 0; j < 2 * N - (2 * i + 1); j++)

        {

            printf("*");

        }

        for (int j = 0; j < i; j++)

        {

            printf(" ");

        }
```

```c
        printf("\n");

    }

}


int main()

{

    int N;

    scanf("%d", &N);

    if(N<0){

            printf("pattern not possible");

    }

    erect_pyramid(N);

    inverted_pyramid(N);

    return 0;

}
```

Testcases:

Testcase- 1:

Input: 'N' = 3

Output:

```
  *
 ***
*****
*****
 ***
  *
```

Testcase- 2:

Input: N = 1

Sample Output 2 :

```
*
```

```
*
```

Testcase- 3:

Input: N= 2

Output:
```
  *
***
***
  *
```

Testcase- 4:

Input: N= 5

Output:
```
      *
    ***
   *****
  *******
*********
*********
  *******
   *****
    ***
      *
```

Testcase- 5:

Input: N= 4

Output:
```
    *
   ***
  *****
*******
*******
  *****
   ***
    *
```

Testcase- 6:

Input: N= -7

Output: pattern not possible

Testcase- 7:

Input: N= 0

Output: